В проекте могут присутствовать: разработчик, продукт менеджер, прожект менеджер, QA, разработчик. В зависимости от масштаба проекта, этих людей может быть больше , они объединяются в команды внутри проекта. Либо кто то из команды, если проект небольшой, может совмещать обязанности (Пример: Разработчик-QA) Но это так себе варианты, всё-таки каждый должен делать своё дело=-) . Но всё-таки имеет место быть.

1. Честно говоря, думал, что вход в профессию через тестирование будет лёгким. Но как всегда есть свои особенности, нюансы. Понимаю сейчас, что хоть только и начали изучать дисциплину, мне очень она импонирует. Отвечать в какой-то степени за готовый продукт это круто. Очень понравился язык программирования Python, сейчас допом изучаю его. Что бы пробовать писать в дальнейшем авто тесты (Лень – двигатель прогресса=-)).
2. Я довольно общителен, открыт (иногда из – за этого страдаю, но это нюансы)) , мне нравится изучать что то новое, теперь понимаю что нравиться развиваться. Как для QA инженера важно изучить проект со всех сторон, нравиться докапаться до сути. Иногда лень одолевает ) , но как пошёл учиться понял, что главное начать! Дальше само пойдет.
3. Думаю хорошие Soft skills + 70% Хардов. Вот субъективный рецепт))). Если серьёзно то знание основ непременно! Чек листы, Тест кейсы, знание TMS-ок(популярных), Язык программирования(по крайней мере чутка понимать что написано в коде, отсюда вытекает – код ревю.), методы тестирования (функциональное, нефункциональное и т.п.),техники тест дизайна. Понимание работы клиент серверной архитектуры, отсюда тестирование API. Не забывать про ТЗ, то вообще основа! Ну а если его нет, то на опыт опираться которого пока нет))). Будет полезно знание операционных систем, мобильных операционок, если на их направление потянет, ааааа и БД!(реляционки - SQL и нереляционки - NoSQL) Соответственно знать язык запросов к БД – шкам. Возможно я конечно загнул знания для первой работы, но это всё мне хочется изучить .
4. Думаю основные задачи это Изучение ТЗ. Написание Тест Кейсов или Чек листов (как показывает практика на проекте выбирают что то одно). Проведение самих тестов, общение с разрабами по косякам. Обсуждения в команде задач, моментов , нюансов.
5. -Функциональность: Для чего продукт задуман , те функции и должен выполнять. В се остальное доп. Затраты по времени и деньгам. Пылесос должен должен всасывать пыль, а не сушить волосы. Хотя…..)))

-Эффективность: Система не должна кушать много ресурсов, чем ей требуется в разные промежутки времени.

-Надёжность: Система должна быстро восстанавливаться после сбоя оборудования.

-Удобство использования: К примеру узнаваемость интерфейса на IOS. Интуитивно понятный интерфейс, быстрота работы (Кстати над интерфейсом первого Айфона довольно сильно потели инженеры Apple, причём в тайне от Стива Джобса=-))

- Поддерживаемость: На сколько продукт можно качественно поддерживать, не покрашится ли после первого же обновления или правки маленького бага. На сколько продукт можно модифицировать не привнеся в него дефектов или багов.

- Переносимость: На сколько легко продукт может переносится с одного окружения в другое. (Работа в разных операционках например)

- Безопасность: Пример конфиденциальность данных в банковских приложениях, как продукт будет защищать наши данные.

-Совместимость: Пример передача картинок из браузера в вацап или другую соцсеть. Правильно ли передаёт, не мешает ли это работе самого браузера. (Думаю так)